Farmer Query

Smiket Barodia-2018A7PS0231H

f20180231@hyderabad.bits-pilani.ac.in

Hritik Singh Kushwah-2018A7PS0323H

f20180323@hyderabad.bits-pilani.ac.in

Akshit-2018A7PS0187H

f201820187@hyderabad.bits-pilani.ac.in

Department of CSIS, Bits-Pilani Hyderabad Campus

***Abstract*—Farmers, from different regions of India, face a huge spectrum of atrocities. The Government of India has shown it's concern by laying the foundations of Kisan Call Centres (KCC) countrywide. KCC registers the queries submitted by the farmer. There is a need to analyze such data so that effective utilization of data can be done in future. Every query has following Attributes: Sector, Category, Crops, Query- Type, State Name, District Name, Block Name and Submission time. In this paper we will discuss how we can analyze the key features of the dataset from various states where agriculture is prominent. We will be using different Data Mining techniques to achieve the task.**

Keywords—Farmers, Query, Data, Processing, Techniques, Feature Selection, PCA, Cramer's V, Apriori, K-Modes Clustering, AEVF, AVF, Z-Score, Outliers.

I. INTRODUCTION

The Data is about the Queries Farmers have registered at Kisan Call Centre (KCC). To resolve Farmer's query, we are going to analyze the data set and answer some fundamental questions like how should local bodies enact to resolve Farmer's concern, which crops are not suitable for particular regions and get some insights on the agricultural domain of the country. To analyze the data, we have used Python 3.7, IDE- Jupyter Notebook, Pandas, sklearn and various other Python Libraries.

II. SELECTING DATA SET

We fetched datasets of the states-Haryana, Gujarat, Jammu-Kashmir, Tamil Nadu, Uttar Pradesh, Rajasthan- in .csv format from data.gov.in, available under Famer Query section. The database is sorted in State-District-Block format, grouped seasonally.

III. DATA PREPROCESSING

The preliminary step in Data Mining is the pre-processing of data, which makes application of Machine Learning Algorithms feasible. Data Pre- processing involves following steps:

A. Data Cleaning

All the NULL entries or missing entries in the dataset were by-default assigned '0'. First, we replaced all the '0' with 'nan' and then dropped all 'nan' values in the dataset. Refer to the Code below: Df. replace('0',np.nan,inplace=True) Df.dropna(axis=0,inplace=True)

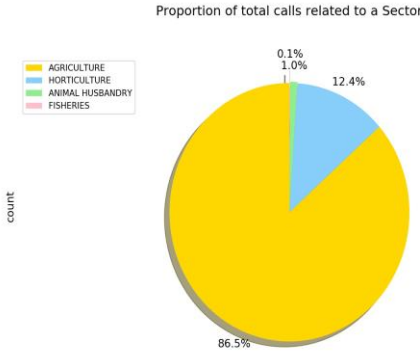
B. Data Visualization

Libraries used for Visualization

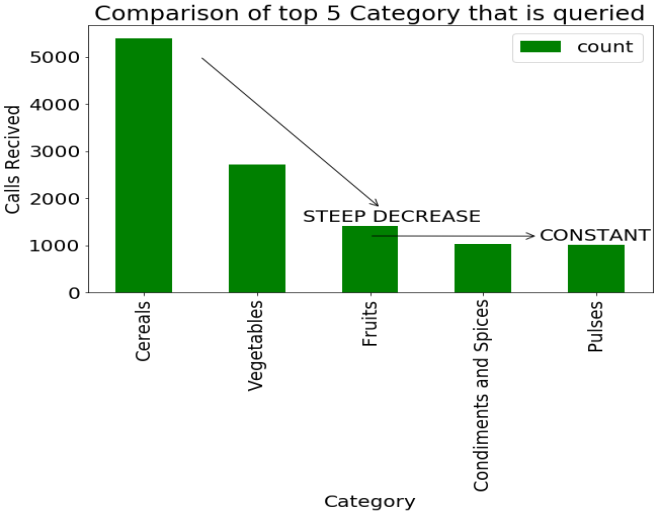
- Matplotlib
- Seaborn
- Folium

Folium library is used to display all the states that have been chosen for analysis on graphs in India.

PLOTS:

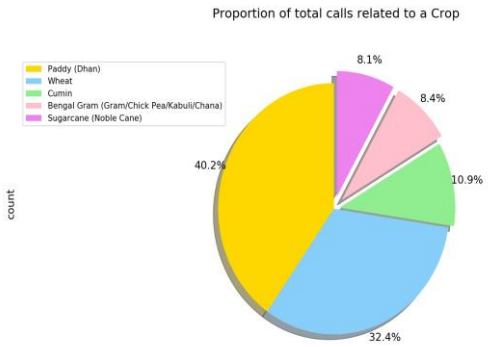


Queries related to Agricultural sector are dominant whereas Animal Husbandry and Fisheries sector queries are insignificant.

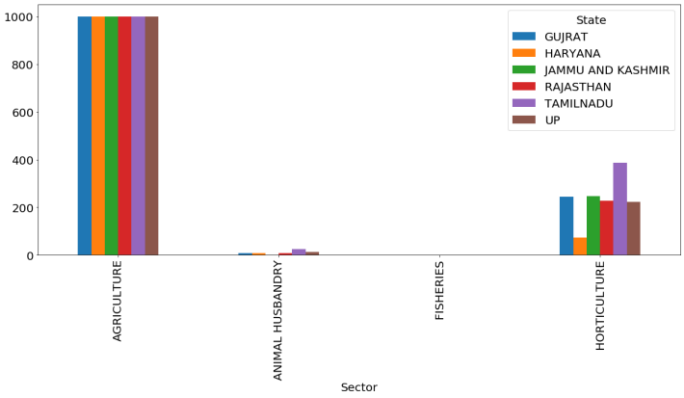


In the above plot, we omitted the Others value of Category Attribute to get better understanding of particular Categories.

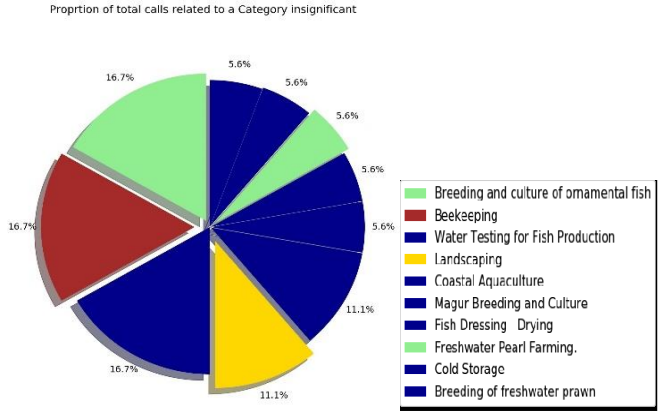
Queries concerning Vegetables and Cereals (count>2500) are high as compared to other categories, maintaining a constant count of 1000.



Calls concerning Paddy and Wheat received were frequent and makes up the majority of the calls (70%).



The above graph (pivot graph) is a graphical representation of Sector v/s State Name Attributes after performing normalization on query count. This shows normalized query calls related to each sector of each state in which Agriculture is taken reference point since all states have max number of calls related to it as compared other sectors. Tamil Nadu outstands in every sector as compared to other states.



The above pie plot represents those categories which received Minimum calls for issues. These categories are generally associated with Fisheries Sector (Represented in Blue) and ornament industry(Represented in Green).

C. Data Transformation

As the entire data is categorical and that to nominal. Data has to be transformed to make it fit for algorithms. One-Hot encoding has been done for applying PCA and other Algorithms which cannot be directly applied to categorical data. But applying this transformation sometimes creates issues as the column size increases to 500 which is not easy to handle.
So as an alternative Label Encoding has been done wherever applicable (where there is no comparison between the values). In Outlier analysis z-normalization was applied after Label Encoding.

D. Data Reduction

For data reduction we applied two strategies i.e. PCA and Attribute Subset Selection. While applying these strategies our Target Variable is QueryType.

1) Applying PCA

For applying PCA for categorical data, we first apply One-Hot Encoding to all the categories and then standardize the data. PCA takes a d dimension input matrix and maps it to k dimension matrix (where k<<d).

PCA
Input: *[R x d] matrix with dimension d*
Output: *[R x k] matrix where k<<d*
Algorithm:
1.Standardizing the d-dimension dataset
2.Making a Covariance matrix
3.Decompose the matrix into its eigenvectors and eigenvalues
4.Selecting top k eigenvalues (as they will contain the largest variance)
5.Mapping d-dimension into k-dimension using a W matrix formed by the eigenvalues
After One-hot Encoding the data, we got 530 columns and after applying the above algorithm we reduced them to 2 columns namely Principal Component 1 and Principal Component 2.
These Components were plotted using Scatter-Plot

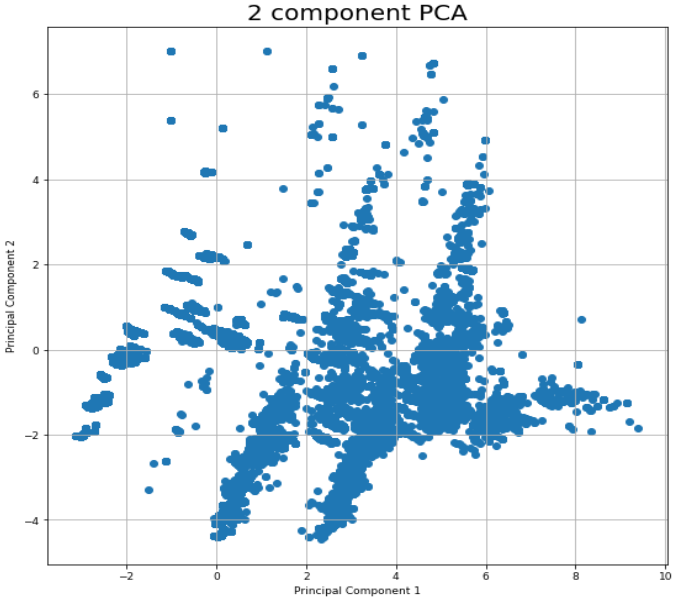


fig.1

But the pca.explained_ratio which tells about how much data is contained in these components were 0.014(PC1) and 0.010(PC2) therefore total ratio becomes 0.024 which is very low and thus cannot be accepted.
Now we will plot a step curve which indicates how many components are needed to be preserved to get Desired explained ratio that is 90%. (See fig.2 for step Curve). So, to get at least 90% of the total data or total ratio =0.9 we found that we will need at least 369 columns. But reduction from 530 columns into 369 was not effective.

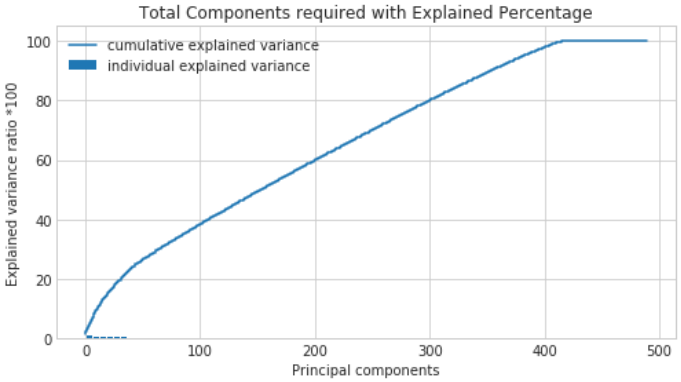


fig.2

The reason for such a low pca.explained_ratio is the less correlation between the categories and the attributes of the dataset after applying one-Hot Encoding.
So, to resolve this issue we will use Attribute Subset Selection for data reduction.

2) Attribute Subset Selection

For Attribute subset selection our objective function will be the correlation of the attribute with our Target variable that is QueryType.
Now correlation for categorical variables is not defined as the usual manner. Formula for correlation between two variables is:-

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

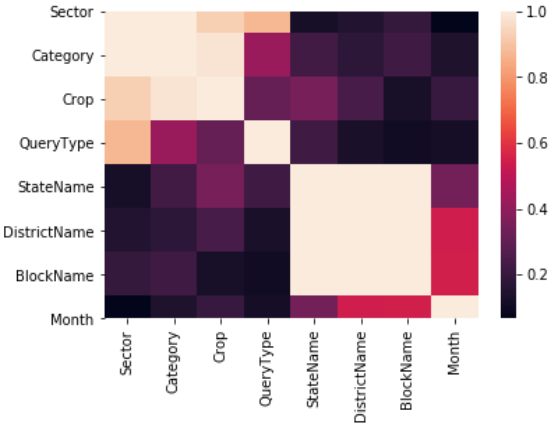
But for categorical data this formula is meaningless.
So, to find correlation between categorical values we used Cramer's V. This method uses Pearson's Chi-squared statistic. This test is helpful in finding the dependency between the two variables. Cramer's V gives output from 0 to 1 where 0 shows no dependency and 1 shows full dependency.
For this method both the variables are converted into a contingency matrix A. In matrix, 'ni,j' denotes the number of times when categories from each variable occur together. Then

$$\chi^2 = \sum_{i,j} \frac{(n_{ij} - \frac{n_{i.}n_{.j}}{n})^2}{\frac{n_{i.}n_{.j}}{n}}$$

And final value V is calculated using the formula.

$$V = \sqrt{\frac{\varphi^2}{\min(k-1, r-1)}} = \sqrt{\frac{\chi^2/n}{\min(k-1, r-1)}}$$

This has been implemented in the python code to obtain the given heat-map of association.



This heat-map clearly shows the association between each attribute. For our target variable the attribute Sector (0.87) and Category (0.5) holds the highest association.
Now each attribute is chosen if it's association with the target variable QueryType is greater than 0.2.
After applying this method, we filtered attributes ('BlockName, DistrictName and Month) from our dataset.

IV. DATA ANALYSIS

Association Analysis

1) Description of Algorithm

Association analysis can be very helpful for our dataset as it can construct insightful rules. For this analysis Attributes Crop, QueryType and StateName are taken into consideration. The rules generated will give us information about the relation between these attributes.
So, Transaction List and itemset was created from the data frame and apriori algorithm was applied to them with different Support for different Cases but mostly taken 0.2 for most cases while the minConfidence is 0.2.
Apriori
Input: *DataFrame of Transaction, minSupport, minConfidence*
Output: *k-frequent itemset*
Algorithm:
1. F1 - Frequent 1- itemset are taken and crossed with itself only if the prefix of both the sets are the same.
F1xF1 = C2
2. Now from Candidate Set C2 sets are pruned if they have infrequent subsets (following apriori pruning rule) thus creating F2.
3.These process goes on repeating till Fk ={Null}.
4. So the Fk-1 set is used to generate rules.
Rule Generation:
5. For every subset P in Fk-1 the rule P -> (Fk-1 -P) can be generated if it satisfies the given confidence.

The transaction List is created by treating each row of the data frame as 1 transaction and itemset are the unique categories of each attribute (E.g. {{paddy}, {Government Schemes}, {Wheat}, etc})

2) Results Obtained and Conclusion drawn

Different rules generated through this algorithm and their corresponding analysis:
1.('HARYANA') ===> ('Weather', 'Others'); ;('RAJASTHAN',) ===> ('Weather', 'Others'); ;('JAMMU AND KASHMIR',) ===> ('Weather', 'Others')

Note: - The value 'Others' for Attribute Crop denotes those crops for which information is not given and can be treated as a general entity for this category.

So, the state of Haryana and Rajasthan generally suffer from Weather issues for most of its Crops.
4.('Paddy (Dhan)', 'Plant Protection') ==>('HARYANA')
The Crop of paddy suffers issue of plant protection in Haryana.

5.('Paddy (Dhan)',) ===> ('HARYANA',); ('Wheat',) ===> ('HARYANA')
These rules indicate that most of the calls about Paddy and Wheat are from Haryana from all the states considered.

Rules between QueryType and Crops in specific State using apriori:

1. Jammu & Kashmir:
   ('Dairy Production',) ==> ('Bovine (Cow, Buffalo)',)
   ('Mango',) ===> ('Nutrient Management',)
   ('Paddy (Dhan)',) ===> ('Plant Protection',)
   Now dairy production in J&K is generally done by Bovine.
   Mangoes suffers from Nutrient Management issue in J&K

2. Tamil Nadu
   ('Paddy (Dhan)',) ===> ('Fertilizer Use and Availability', 'Nutrient Management', 'Plant Protection')
   This indicates Paddy in Tamil Nadu suffers from given issues.

3. Gujarat
   ('Brinjal',) ===> ('Plant Protection',)
   ('Cotton (Kapas)',) ===> ('Plant Protection',)
   ('Chillies',) ===> ('Plant Protection',)
   Brinjal, Cotton and Chillies all suffer from plant Protection In Gujarat

4. Rajasthan
   ('Bengal Gram,) ==> ('Plant Protection',)
   ('Cumin',) ===> ('Plant Protection',)
   ('Wheat',) ===> ('Nutrient Management',)
   ('Mustard',) ===> ('Plant Protection',)
   Bengal Gram and Mustard suffer from Plant Protection issues whereas Wheat suffers from Nutrient Management issues.

The Crops which are mostly grown in these regions can be extracted from these rules as they are the itemset involved. For eg it's obvious that in Gujarat Brinjal and Cotton are grown as major crops.

All the rules, nation as well as state-wise can be found in the notebook on the given GitHub repo.

B. Cluster Analysis

A very few techniques are available for clustering of nominal data. Some of them are entropy bases such as AEVF. K-Modes is an eminent algorithm for clustering data set with categorical attributes. It uses Hamming Distance as distance metric to measure dissimilarity between categorical objects.

1) K-Modes-Clustering

$$P(W,Q)=\sum_{l=1}^{k}\sum_{i=1}^{n} w_{il} d_{sim}(x_i,q_l) \quad (1)$$

$$d_{sim}(x_i,q_l)=\sum_{j=1}^{m}\delta(x_{ij},z_{lj}) \quad (2)$$

where, $\delta(x_{ij}, q_{lj})$ is calculated using the following Eq.(3)

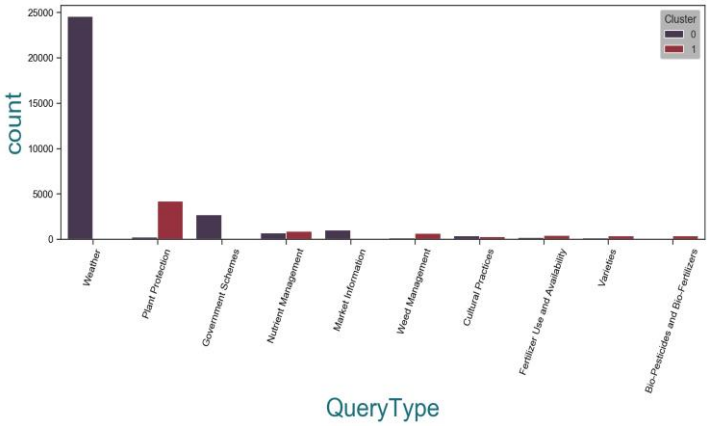
$$\delta(x_{ij},z_{lj})=\begin{cases}1 & if\ x_{ij}=z_{lj}\\ 0 & if\ x_{ij}\neq z_{lj}\end{cases} \quad (3)$$

Input: Data Frame, number of clusters k
Output: k clusters
Algorithm:

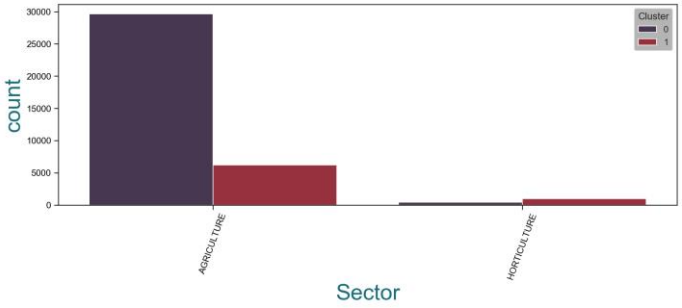
1. k objects are chosen as modes depending upon initialization method.
2. Find the matching dissimilarity between the each K-initial cluster modes and each data object using the Eq(2).
3. Evaluate the fitness using the Eq.(1)
4. Find the minimum mode values in each data object i.e. finding the objects nearest to the initial cluster modes.
5. Assign the data objects to the nearest cluster centroid modes.
6. Update the modes by applying the frequency-based method on newly formed clusters.
7. Recalculate the similarity between the data objects and the updated modes.
8. Repeat the step 4 and step 5 until no changes in the cluster ship of data objects.

Before Applying K-Modes, we performed Ensemble Selection and selected top 10 values from Attributes QueryType, Category and Crop so that we could see the effect of clustering properly.

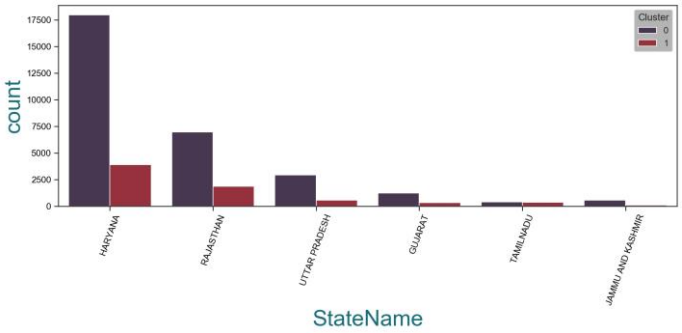
Then we applied Label Encoding on the data to convert Categorical Attributes to Numerical.
To find the optimal number of clusters, we used the Elbow method, indicating us k=2 for our dataset.



Cluster 0 consisted mainly of the Weather-related query whereas cluster 1 had Plant Protection related query the most.



Cluster 0 was crowded with Agriculture related queries. Horticulture queries were significant in cluster 1.



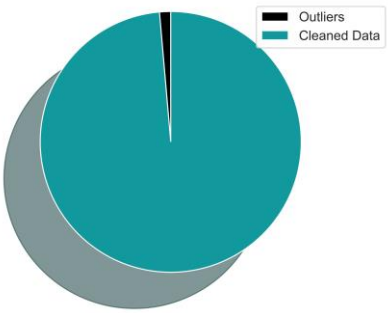
Around 17,500 queries from Haryana were part of the bigger Cluster 0. States of Rajasthan, Uttar Pradesh and Gujarat followed the same fashion. Whereas queries from Tamil Nadu were equally residing in both the Clusters.

C. Outlier Analysis

The methods for numerical data cannot be directly applied to categorical data, because it needs mappings of categorical attributes into numerical attributes. The AVF (Attribute Value Frequency) method uses frequency data and it is similar to distance based method for numerical data. AEVF(Automated Entropy Value Frequency) used entropy change to determine the degree of outliers. Outliers make large changes in entropy.

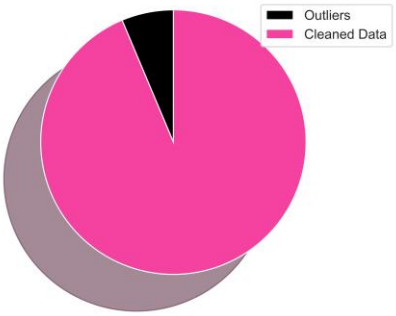
1) Using Clustering

According to the dimensions of our dataset, we used K-Modes Clustering Algorithm to make 40.
Clusters out of which we pruned the clusters containing objects less than 100. The proportion is shown below.



2) Using Z-Score

Using the Z-Score function from scipy.stats we selected the objects having z value less than 3 and rest were marked as outliers. The proportion is shown below.



V. CONCLUSION

Datasets from different states and Categories of Crops was analyzed using various techniques. As dataset was categorical in nature Data visualization served as one of the major assets to get critical and valuable information. Plots were generated between each attribute and carefully analyzed.
New method for data reduction of categorical data is implemented rather than the conventional PCA. For analysis,
The information obtained from association analysis (i.e. Apriori) provided us with insights about the relation between the target variable Query Type and different important factors like Crop.
Clustering was implemented for categorical data using k-modes which takes ‘mode’ of the attribute as the deciding criteria.
Finally, Outlier analysis has been done through clustering and by normalizing the data with z-score separately. Comparisons were drawn between the two and thus implementing the better one.
In conclusion, different data mining techniques were successfully implemented and insightful conclusions were drawn from them.

References

1. Huang, Z.: Extensions to the K-Means algorithm for clustering large data sets with categorical values. Data Mining Knowledge Discovery 2(3), 283–304 (1998)
2. Cao, F., Liang, J., Bai, L.: A new initialization method for categorical data clustering. Expert Systems with Applications 36, 10223–10228 (2009)
3. Fast Entropy Attribute Value Frequency Algorithm to Detect Outliers for Categorical Data Kang-Mo Jung-Kunsan National University
4. Initialization of K-modes clustering using outlier detection techniques-Feng Jiang,Guzhu Li, Junwei Du, Yuefei Sui
5. seaborn.org
6. https://towardsdatascience.com/the-search-for-categorical-correlation-a1cf7f1888c9
7. python.org
8. https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02
9. dython/dython at master · shakedzy/dythonhttps://github.com/asaini/Apriori/blob/804020f9547b7ac55e0101d80522d60fe294c18e/apriori.py#L59
10. http://sebastianraschka.com/Articles/2015_pca_in_3_steps.html
11. https://stats.stackexchange.com/questions/443878/when-to-use-theils-u-and-cramers-v-the-danger-of-symmetrical-dat

GitHub Repository -https://github.com/akshitkh47612/Data-Analysis-of-Farmer-query/import